

Retrieve, Reason, Verify: A Modular Multi-Model System for DocSem

Pranavi Tadivalasa

Oracle

pranavi.tadivalasa@oracle.com

Abstract

DocSem requires systems to answer quantitative questions over document images while also returning the evidence supporting each prediction. We present *Retrieve, Reason, Verify*, a modular system that separates document transcription, evidence localization, numerical reasoning, deterministic execution, and selective multi-model verification. Under controlled evidence grounding on 217 validation instances, Qwen3.8-Max achieves 98.62% Joint Exact Accuracy. Combining Qwen3.8-Max and GPT-5.6 Sol as independent solvers, with Gemini 3.1 Pro invoked only when their numerical predictions disagree, improves validation performance to 99.08%. On the 1,730-instance hidden test set, which presents more challenging visual and grounding conditions, including degraded document quality, non-uniform evidence identifiers, and multiple plausible evidence passages, the complete end-to-end pipeline achieves 32.7% Joint Exact Accuracy. A manual audit of 50 hidden-test instances shows that the system-selected evidence differs from the manually identified evidence in 34 cases, whereas only two cases produce different numerical answers when the evidence agrees. These observations suggest that, within the audited sample, errors are concentrated more strongly in document grounding than in downstream numerical computation once the relevant evidence has been established.

1 Introduction

Quantitative question answering over document images requires more than extracting text and performing arithmetic. Before numerical reasoning can succeed, a system must identify the portion of the document containing the relevant quantitative scenario, recover that content accurately, preserve its associated evidence identifier, and determine the computation required by the query. Errors introduced at any of these stages can invalidate an otherwise correct numerical solution.

The DocSem shared task¹ brings these challenges together in a single evaluation setting. Systems receive a PDF document and a paraphrased query and must return both a numerical answer and the visible document block identifiers supporting that prediction. Queries may refer to the target scenario only indirectly, while documents can contain multiple numerical passages, repeated entities, visually degraded content, and administrative or contextual text with substantial lexical overlap. Consequently, lexical similarity alone is often insufficient for identifying the correct evidence. A successful system must recover the document structure, distinguish semantically relevant passages from plausible distractors, and preserve the correspondence between the selected evidence and the final numerical prediction.

We address this problem with *Retrieve, Reason, Verify*, a modular pipeline organized around a separation between document grounding and numerical computation. The grounding stage reconstructs the document into identifier-preserving text blocks and localizes the passage containing the complete quantitative scenario. The reasoning stage converts the grounded problem into an executable arithmetic expression, which is evaluated deterministically. Multiple reasoning models are then combined selectively, with an additional verifier invoked only when their numerical predictions disagree. This modular design enables numerical reasoning to be evaluated independently of document-grounding quality and facilitates stage-wise analysis of end-to-end errors.

Our experiments reveal a substantial difference between reasoning under fixed evidence and complete end-to-end document processing. When the relevant evidence is held constant, the strongest evaluated reasoning models solve nearly all validation instances correctly, and selective multi-

¹<https://docinsights-workshop.github.io/docinsights-2026/shared-task/>

model adjudication further reduces residual errors. End-to-end performance, however, is substantially lower when the system must first recover the relevant evidence from the document. A manual audit of hidden-test predictions further indicates that, within the inspected sample, discrepancies between system predictions and manually derived solutions occur substantially more often in evidence selection than in numerical reasoning when the same evidence is identified. This distinction motivates treating document grounding and numerical reasoning as separate sources of error rather than evaluating the system as a single opaque prediction process.

Our main contributions are:

1. We propose *Retrieve, Reason, Verify*, a modular architecture for quantitative document question answering that explicitly separates document grounding, numerical reasoning, deterministic execution, and selective verification.
2. We evaluate multiple reasoning models under controlled evidence grounding, enabling quantitative reasoning performance to be measured independently of document-localization errors.
3. We provide a manual analysis of the complete DocSem pipeline that distinguishes discrepancies in evidence recovery from discrepancies in downstream numerical reasoning and identifies grounding as a major source of error within the audited hidden-test sample.

2 Related Work

2.1 Document Question Answering

Document question answering requires models to jointly exploit textual content and the visual structure of document images. DocVQA established a benchmark for visual question answering over documents, emphasizing the importance of reasoning over document content beyond plain-text extraction (Mathew et al., 2021). Subsequent work has increasingly incorporated multimodal representations of text, layout, and visual appearance. LayoutLMv3 jointly models textual and visual information through unified multimodal pre-training (Huang et al., 2022), while Donut adopts an OCR-free formulation that maps document images directly to structured outputs (Kim et al., 2022).

These approaches primarily address the recovery and interpretation of document content. DocSem introduces an additional requirement of evidence attribution, whereby the predicted answer

must be associated with the correct evidence identifier. Document processing must therefore preserve the correspondence between textual content and its original document-level identifier. Our system maintains this association during reconstruction by representing the document as identifier-preserving text blocks prior to evidence localization and numerical reasoning.

2.2 Numerical Reasoning and Program Execution

Numerical reasoning over documents has been studied in settings that combine evidence retrieval with multi-step computation. FinQA evaluates numerical question answering over financial reports, where systems must identify relevant facts and compose executable reasoning programs (Chen et al., 2021). TAT-QA extends this setting to hybrid evidence distributed across textual passages and tables (Zhu et al., 2021). These benchmarks demonstrate that numerical question answering depends jointly on accurate evidence selection and reliable execution of the required computation.

A related line of work separates language-based reasoning from numerical execution. PAL prompts language models to generate executable programs and delegates the resulting computation to a Python interpreter (Gao et al., 2023). Program of Thoughts similarly externalizes arithmetic operations into executable reasoning steps rather than relying on implicit calculation within the language model (Chen et al., 2023). Our system follows this general paradigm by deriving a constrained arithmetic expression from localized evidence and evaluating it deterministically. This separation enables errors arising from evidence selection, reasoning formulation, and numerical execution to be examined independently.

2.3 Reasoning Robustness and Verification

The reliability of numerical reasoning can be affected by irrelevant, distracting, or semantically similar context. Shi et al. demonstrate that irrelevant information can substantially degrade mathematical reasoning performance, particularly when distractors appear plausible with respect to the query (Shi et al., 2023). GSM-SEM further examines the sensitivity of mathematical reasoning systems to meaning-preserving semantic variation (Singh et al., 2026). These findings are relevant to document-level quantitative reasoning, where multiple passages may contain overlapping entities or

numerical information even though only one provides the complete evidence required to answer the query.

Prior work has also explored inference-time aggregation as a mechanism for improving reasoning reliability. Self-consistency samples multiple reasoning trajectories from the same model and selects the final answer based on agreement across the resulting solutions (Wang et al., 2023). In contrast, our system obtains independent predictions from distinct reasoning models and performs additional verification only when their numerical outputs disagree. This disagreement-triggered strategy concentrates additional reasoning effort on uncertain instances while avoiding an additional verification step when the primary solvers already agree.

3 System Description

Given a PDF document and a query, the system predicts a numerical answer together with the identifier of the supporting evidence. Figure 1 provides an overview of the architecture. The pipeline first reconstructs the document into identifier-preserving text blocks and then localizes the passage containing the relevant quantitative scenario. From the localized evidence, a reasoning model derives an executable arithmetic expression, which is evaluated deterministically. Independent solvers operate over the same grounded input, and disagreement cases are routed to a separate verifier.

Throughout the paper, we use *grounding* to refer collectively to document reconstruction and evidence localization. Transcription recovers textual content from the document image, reconstruction associates that content with document-level identifiers, localization selects the evidence required for the target problem, and reasoning derives the corresponding computation.

3.1 Core Pipeline

The core pipeline is designed to separate document grounding from numerical reasoning while preserving the correspondence between the selected evidence and the final prediction.

Document Reconstruction Each PDF page is rendered with PyMuPDF at 150–200 DPI and transcribed using PaddleOCR with PP-OCRv6 (PaddlePaddle, 2026). Because the validation documents are visually high quality, this rendering resolution is sufficient to preserve textual and numerical content without requiring higher-resolution pro-

cessing. The OCR output is segmented into text blocks while retaining the evidence identifier associated with each passage. The resulting representation contains both the reconstructed document content and the set of valid identifiers required by the subsequent localization stage.

Evidence Localization Given the query, reconstructed text blocks, and valid evidence identifiers, Qwen2.5-7B-Instruct identifies the smallest evidence set containing the complete quantitative scenario. The model additionally extracts the mathematical question expressed within the selected passage. Separating passage selection from question extraction is useful when the user query refers to the target scenario indirectly or does not reproduce the wording of the underlying quantitative problem.

Quantitative Reasoning The reasoning model receives the localized evidence, extracted mathematical question, and original user query. It first identifies the target quantity and the numerical values, units, and relationships required for the computation and subsequently produces a Python-compatible arithmetic expression. The reasoning prompt explicitly accounts for unit conversion, percentage interpretation, comparison direction, omitted or duplicated quantities, and irrelevant numerical information. Constraining the solver to produce an executable expression separates semantic reasoning from the final arithmetic evaluation.

Deterministic Execution Following prior work on program-aided numerical reasoning (Gao et al., 2023; Chen et al., 2023), the generated expression is evaluated using a restricted Abstract Syntax Tree (AST)-based Python executor. The executor permits standard arithmetic operators, unary operations, and a predefined set of numerical functions while rejecting unsupported constructs. Malformed or unparsable expressions trigger up to two structured-output recovery attempts before the instance is treated as an execution failure. This design avoids relying on the language model to perform the final arithmetic operation internally.

Selective Multi-Model Verification Two reasoning models independently solve the same grounded problem. Their numerical outputs are normalized to a common representation and compared directly. Predictions with identical normalized values are accepted without additional verification. When the

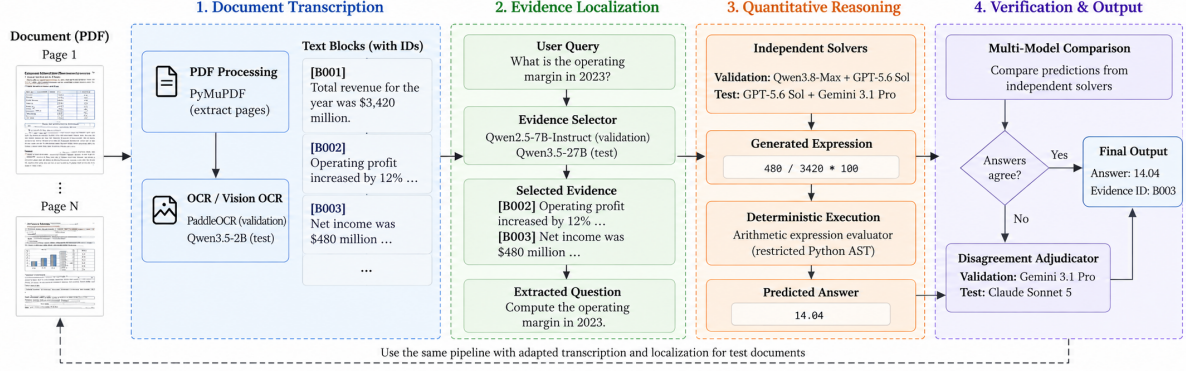


Figure 1: Overview of the proposed *Retrieve, Reason, Verify* pipeline. The document is reconstructed while preserving evidence identifiers, the relevant quantitative passage is localized, independent reasoning models derive executable computations, and disagreement cases are routed to a separate verifier.

predictions disagree, a separate verifier receives the localized evidence and both candidate derivations and independently reconstructs the required computation. Verification is therefore restricted to disagreement cases, concentrating additional reasoning capacity on instances for which the primary solvers do not reach the same result.

3.2 Hidden-Test Adaptations

The hidden-test setting presents additional challenges at the document-processing stage. Although reference answers and evidence annotations are unavailable, the input documents exhibit visual and formatting characteristics that differ from the validation data, including degraded text quality, compression artifacts, and non-uniform evidence identifiers. We therefore adapt the transcription and localization components using only observable properties of the unlabeled test documents. The interfaces among grounding, reasoning, deterministic execution, and verification remain unchanged.

Visual Transcription For hidden-test documents, we replace the OCR-based transcription used during validation with Qwen3.5-2B. Qualitative inspection of the unlabeled hidden-test documents indicated that conventional OCR pipelines, including PaddleOCR and Tesseract, were less reliable for low-contrast text and irregular identifier strings. We therefore use a vision-language model for transcription and instruct it to reproduce visible document content without normalizing capitalization, punctuation, numerical values, or identifier strings.

Format-Agnostic Block Reconstruction Unlike the validation data, hidden-test evidence identifiers

do not follow a consistent prefix, length, or character pattern. The reconstruction stage therefore avoids imposing a predefined identifier format. Instead, the complete string preceding the first colon of each detected block is retained as the evidence identifier. Visually similar characters are preserved without normalization in order to maintain the document representation as faithfully as possible.

Evidence Localization The hidden-test localizer receives the reconstructed document and identifies the passage containing the target quantitative scenario. The predicted identifier is subsequently checked against the identifier set extracted during reconstruction. If the predicted identifier is invalid, localization is repeated while restricting the model to valid candidates, for at most two additional attempts. This constraint reduces identifier-formatting errors without modifying the underlying document content.

Reasoning and Adjudication Once evidence has been localized, two independent solvers derive the numerical solution from the same grounded input. Predictions with identical normalized numerical values are accepted directly. For disagreement cases, a separate adjudicator receives the localized evidence together with both candidate derivations and independently reconstructs the final computation. The hidden-test system therefore retains the same disagreement-driven verification strategy while adapting the transcription and localization components to the observed test-time document conditions.

4 Experimental Setup

4.1 Evaluation Protocol

Prompt templates, document-processing rules, and pipeline design choices are developed using the 908 labeled DocSem training instances. The 217 validation instances are held out from this development process and subsequently used for comparison of quantitative reasoning configurations. To decouple numerical reasoning from document-grounding errors, every validation solver receives the same localized evidence and extracted mathematical question. We refer to this setting as *controlled evidence grounding*.

The hidden test set contains 1,730 instances and is evaluated through the official shared-task framework. In contrast to controlled validation, hidden-test evaluation measures the complete end-to-end pipeline, including document transcription, reconstruction, evidence localization, numerical reasoning, and evidence prediction.

The two settings therefore evaluate complementary aspects of the system. Controlled validation measures reasoning performance conditional on fixed grounding, whereas hidden-test evaluation measures the effectiveness of the complete document-processing and reasoning pipeline.

4.2 Models and Configurations

We evaluate Grok 4.6 Fast, Gemini 3.1 Pro, Qwen3.7-Plus, Qwen3.8-Max, DeepSeek-R1-Distill-Qwen-32B, DeepSeek-R1, GPT-5.6 Sol, Claude Sonnet 5, and Kimi K2.5 Instant as quantitative reasoning models. Complete model references, stage assignments, and inference configurations are provided in Appendix A.

Grok 4.6 Fast is evaluated with both the initial and refined reasoning prompts, while Claude Sonnet 5 is evaluated under medium and higher reasoning-effort configurations. All remaining models use the refined structured reasoning prompt. The validation set is used to compare reasoning-model configurations after prompt and pipeline development on the training data.

The strongest validation configuration combines Qwen3.8-Max and GPT-5.6 Sol as independent solvers, with Gemini 3.1 Pro invoked only when their numerical predictions disagree. For the final hidden-test system, GPT-5.6 Sol and Gemini 3.1 Pro serve as the independent solvers, while Claude Sonnet 5 performs disagreement adjudication. Complete model and inference details are

provided in Appendix A.

4.3 Prompting Strategy

Evidence localization and numerical reasoning are handled using separate prompt templates. The localization prompt instructs the model to identify the minimal evidence required to reconstruct the complete quantitative scenario rather than selecting passages solely on the basis of lexical similarity to the query. The localizer is explicitly restricted to evidence selection and does not perform the numerical computation.

The reasoning prompt requires the solver to identify the target quantity and the numerical relations necessary to answer the question before producing an executable arithmetic expression. The prompt additionally checks for unit consistency, percentage interpretation, comparison direction, duplicated quantities, and unsupported numerical values.

The adjudication prompt is designed to evaluate the underlying computation rather than select an answer according to model identity or majority preference. Given the grounded evidence and competing derivations, the adjudicator independently reconstructs the required calculation and may reject both candidate solutions when neither is supported by the evidence. Full prompt templates are provided in Appendix B.

4.4 Numerical Agreement and Execution

Before solver outputs are compared, numerical answers are converted to a common normalized representation using the same procedure throughout validation and hidden-test inference. Two predictions are considered to agree when their normalized numerical values are identical. Disagreement triggers the additional verification stage described in Section 3.

Expressions produced by the reasoning models are evaluated exclusively through the restricted AST-based executor. Invalid expressions are not evaluated directly and instead follow the structured-output recovery procedure, with at most two recovery attempts. This separation ensures that comparisons across reasoning models primarily reflect differences in problem interpretation and expression construction rather than differences in arithmetic execution.

4.5 Evaluation Metrics

We report the official DocSem metrics: *Answer Exact Accuracy*, *Evidence F1 (macro)*, and *Joint*

Model	Joint Exact Acc. (%)
Grok 4.6 (Fast, initial)	89.86
Qwen3.7-Plus	89.86
DeepSeek-R1-Distill-Qwen-32B	90.78
Kimi K2.5 (Instant)	96.31
Grok 4.6 (Fast, refined)	97.24
Gemini 3.1 Pro	97.24
Claude Sonnet 5 (medium)	97.24
Claude Sonnet 5 (higher)	97.70
GPT-5.6 Sol	98.16
DeepSeek-R1 (671B)	98.16
Qwen3.8-Max	98.62

Table 1: Performance of individual reasoning models on the 217 validation instances under controlled evidence grounding. Evidence is fixed and correct across configurations; consequently, Evidence F1 is 100% and Answer Exact Accuracy equals Joint Exact Accuracy.

Exact Accuracy.

Answer Exact Accuracy measures the proportion of instances for which the normalized predicted answer exactly matches the reference answer. For each task instance, evidence precision and recall are computed using set overlap between the predicted and reference evidence sets, and their harmonic mean gives the instance-level evidence F1. *Evidence F1 (macro)* is obtained by macro-averaging these scores across all instances, allowing partial credit when the predicted evidence set partially overlaps with the reference evidence.

Joint Exact Accuracy requires both the numerical answer and the complete evidence set to be correct for the same instance. We use *Joint Exact Accuracy* as the primary metric because it directly captures the joint requirement of correct numerical reasoning and exact evidence attribution.

5 Results

5.1 Validation Model Comparison

Table 1 reports the performance of individual reasoning models on all 217 validation instances under controlled evidence grounding. Because the same correct evidence is supplied to every solver in this setting, Evidence F1 is 100% across runs, and Answer Exact Accuracy therefore coincides with Joint Exact Accuracy. We consequently report Joint Exact Accuracy as the single summary measure for the validation reasoning comparison.

Qwen3.8-Max achieves the highest single-model Joint Exact Accuracy at 98.62%, corresponding to 214 of 217 validation instances. GPT-5.6 Sol and DeepSeek-R1 each achieve 98.16%, followed by Claude Sonnet 5 with higher reasoning effort at

Configuration	Joint Exact Acc. (%)
Qwen3.8-Max	98.62
GPT-5.6 Sol	98.16
Qwen + GPT + Gemini adjudication	99.08

Table 2: Validation performance of disagreement-based multi-model adjudication under controlled evidence grounding.

97.70%. The relatively small differences among the strongest models indicate that several evaluated solvers achieve comparable performance once the relevant evidence has been fixed.

5.2 Prompting and Reasoning Analysis

Prompt refinement has a substantial effect on Grok 4.6 Fast. The initial prompt achieves 89.86% Joint Exact Accuracy, whereas the refined structured prompt reaches 97.24%, an improvement of 7.38 percentage points.

The refined prompt introduces an explicit decomposition step before expression generation. The solver first identifies the target quantity, relevant values and units, dependencies among quantities, and potential distractors. It then verifies common sources of error, including percentage interpretation, comparison direction, unit conversion, and omitted or duplicated quantities, before producing the executable expression.

The resulting expression is evaluated separately by the restricted Python executor. This design assigns semantic interpretation and computation formulation to the reasoning model while keeping the final arithmetic evaluation deterministic. The improvement observed for Grok 4.6 Fast demonstrates that, in this setting, structured decomposition can substantially improve reasoning performance even when the underlying model and grounded evidence remain unchanged.

5.3 Multi-Model Adjudication

We next examine whether the residual errors of strong individual solvers are complementary. Qwen3.8-Max and GPT-5.6 Sol independently solve each validation instance using identical grounded evidence. Gemini 3.1 Pro is invoked only when their normalized numerical predictions disagree.

The adjudicated configuration achieves 99.08% Joint Exact Accuracy, corresponding to 215 of 217 validation instances and recovering one additional instance relative to the strongest individual solver.

This improvement indicates that the remaining errors of Qwen3.8-Max and GPT-5.6 Sol are not fully overlapping and that disagreement-based verification can resolve a subset of residual reasoning errors.

The improvement is modest in absolute terms, increasing Joint Exact Accuracy by 0.46 percentage points, and disagreement cases require an additional model invocation. Selective adjudication limits this additional inference cost to cases in which the primary solvers disagree, rather than applying verification uniformly to all instances.

5.4 Hidden-Test Performance

The final end-to-end system achieves 32.7% Joint Exact Accuracy on the 1,730-instance hidden test set. The hidden-test evaluation reports only Joint Exact Accuracy; separate Answer Exact Accuracy and Evidence F1 scores are not available for this split.

Unlike the controlled validation setting, this score incorporates errors from visual transcription, block reconstruction, evidence localization, exact identifier recovery, and numerical reasoning. The hidden-test result should therefore not be interpreted as a direct measure of reasoning degradation relative to validation. Instead, it reflects the combined performance of the complete document-processing and reasoning pipeline.

6 Manual Error Analysis

To investigate the sources of hidden-test error, we manually solve 50 instances from the 1,730-instance test set and compare the resulting evidence selections and numerical answers with the predictions produced by our system. For each instance, we inspect the corresponding document passages and determine the relevant quantitative problem, supporting evidence, and numerical solution directly from the document. Numerically equivalent answer representations are treated as agreement. Because this analysis covers only a subset of the hidden test set and relies on manual assessment rather than organizer-provided reference labels, the resulting counts are intended to characterize recurring failure modes rather than estimate the exact error distribution of the complete test set.

6.1 Grounding and Identifier Errors

Manual inspection reveals two recurring challenges in the document-grounding stage.

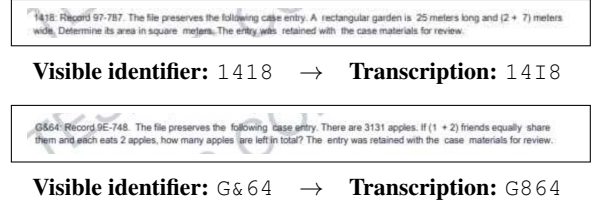


Figure 2: Representative identifier-transcription errors from the hidden test set. Although the surrounding passage remains largely readable, a local character-level transcription error changes the evidence identifier and can therefore affect exact evidence matching.

First, supporting passages are associated with opaque evidence identifiers rather than predictable labels such as b01. These identifiers may contain arbitrary combinations of letters, digits, capitalization, and punctuation, while exact evidence evaluation requires the predicted identifier to match the expected string.

Evidence recovery is therefore sensitive to localized transcription errors. Figure 2 illustrates two representative cases. In both examples, most of the surrounding passage is transcribed correctly, but a single character-level error changes the predicted evidence identifier. Because opaque identifiers provide little semantic redundancy, the surrounding passage often provides insufficient information to reconstruct an incorrectly recognized character.

Second, individual documents frequently contain multiple passages that are superficially relevant to the query. Administrative text may reuse query terms, while separate case descriptions may mention the same entities or contain similar numerical information. Evidence localization therefore requires identifying the passage that fully specifies the target quantitative scenario rather than relying primarily on lexical overlap.

This observation is consistent with prior findings that semantically plausible distractors can affect language-model reasoning (Shi et al., 2023). In the DocSem setting, however, an incorrect localization decision has an additional consequence: it changes the quantitative problem subsequently presented to the reasoning model.

6.2 Evidence and Answer Errors

Table 3 summarizes the relationship between the system predictions and manually derived solutions across the 50 inspected instances.

The system-selected evidence differs from the manually identified evidence in 34 of the 50 in-

System prediction vs. manual solution	Count
Same evidence, same numerical answer	14
Different evidence, same numerical answer	6
Same evidence, different numerical answer	2
Different evidence, different numerical answer	28
Total	50

Table 3: Comparison between system predictions and manually derived solutions for 50 hidden-test instances. Numerically equivalent answer representations are treated as agreement.

spected instances. The largest category consists of 28 cases in which both the evidence and numerical answer differ. Manual inspection reveals a recurring pattern in which the system localizes a different quantitative passage from the one identified during manual analysis and subsequently solves the problem contained in that passage. In these cases, the numerical disagreement appears to originate upstream from evidence localization rather than from arithmetic reasoning over the same evidence.

For example, in one inspected instance, the system selects evidence E77 and produces an answer of 3650, whereas manual analysis identifies evidence B74 and yields 346. The identifiers correspond to different quantitative passages in the same document, illustrating how an incorrect localization decision can determine the numerical problem subsequently presented to the solver.

A further six instances produce numerically equivalent answers despite differences between the system-selected and manually identified evidence. These cases include identifier-transcription differences as well as instances in which distinct candidate passages yield the same numerical result. Such examples demonstrate why answer accuracy alone does not fully characterize performance on a task that jointly evaluates numerical prediction and evidence attribution.

Only two of the 50 inspected instances produce different numerical answers when the system and manual analysis identify the same evidence. Within the audited sample, numerical prediction is therefore substantially more stable once grounding has been fixed. This observation is consistent with the high reasoning accuracy obtained under controlled evidence grounding, although the limited audit size and absence of organizer-provided hidden-test labels prevent conclusions about the exact error distribution of the complete test set.

Taken together, the audit suggests a stage-wise

pattern of error propagation. Evidence localization determines which quantitative problem is presented to the solver, identifier recovery determines whether the supporting evidence is represented correctly for evaluation, and numerical reasoning operates only after these grounding decisions have been made. Within the inspected sample, errors in the upstream grounding stages are substantially more frequent than numerical errors when the relevant evidence has already been identified.

7 Discussion and Future Directions

The manual analysis suggests that further improvements may benefit from targeting the document-grounding stages of the pipeline. In particular, the audited examples expose a coupling between semantic evidence localization and exact identifier recovery: a system may identify the relevant passage while still failing evidence evaluation because the associated identifier is transcribed incorrectly. The following directions could reduce this dependency.

Region-Based Grounding Evidence localization need not depend on first recovering an opaque textual identifier. A system could instead preserve the spatial coordinates of detected text blocks and perform localization directly over document regions. The identifier associated with the selected region would then be recovered only when constructing the final prediction. Such a design would decouple semantic passage selection from identifier transcription and reduce the effect of character-level recognition errors on evidence localization.

Constrained Identifier Recovery The evidence identifiers appearing within a document constitute a finite candidate set. When identifier transcription is uncertain, a predicted string could therefore be reconciled against the observed candidates using character-level similarity, edit distance, or recognition confidence. This correction would operate only over identifiers already present in the document, preserving the requirement that the final output contain the exact canonical identifier expected by the evaluator.

Direct Visual Grounding An alternative is to reduce dependence on full-document transcription altogether. OCR-free document-understanding models such as Donut operate directly on visual document representations (Kim et al., 2022). A future DocSem system could first localize the relevant

page region using a vision-language model and subsequently transcribe only the selected evidence. This approach could limit error propagation from irrelevant or visually degraded regions of the document.

Grounding Verification The current verification stage operates after numerical reasoning, whereas the audit indicates that many observed errors arise earlier in the pipeline. A complementary grounding verifier could assess whether the selected passage contains all quantities, entities, and relationships required by the query before invoking the numerical solver. Disagreement between independently generated localization predictions could provide a natural trigger for this additional verification step.

Overall, these directions suggest shifting part of the verification effort from downstream numerical reasoning toward upstream document grounding. Such changes would preserve the modular structure of *Retrieve, Reason, Verify* while addressing the failure modes most frequently observed in the audited hidden-test examples.

8 Conclusion

We presented *Retrieve, Reason, Verify*, a modular system for document-grounded quantitative question answering that separates document grounding, numerical reasoning, deterministic execution, and selective multi-model verification. Under controlled evidence grounding, the strongest individual solver achieves 98.62% Joint Exact Accuracy, while disagreement-based adjudication improves performance to 99.08%.

The complete end-to-end system achieves 32.7% Joint Exact Accuracy on the 1,730-instance hidden test set. A manual audit of 50 test instances reveals substantially greater error in evidence selection than in numerical reasoning once the evidence is fixed: the system-selected evidence differs from the manually identified evidence in 34 cases, whereas only two cases produce different numerical answers when the evidence agrees.

These findings highlight an important distinction between reasoning accuracy under fixed grounding and reliability in end-to-end document processing. Within the audited sample, numerical reasoning is comparatively stable once the relevant evidence has been established, while transcription, passage localization, and exact identifier recovery introduce substantially more error. The results therefore motivate treating document grounding and numerical

reasoning as separate components when designing and evaluating quantitative document question answering systems.

9 Limitations

Our analysis has several limitations. First, the hidden-test error analysis is based on 50 manually solved instances from a total of 1,730. The resulting counts are intended to identify recurring failure patterns and should not be interpreted as estimates of the exact error distribution over the complete hidden test set.

Second, organizer-provided reference answers and intermediate annotations are unavailable for the hidden test. The audit therefore relies on manual interpretation of the documents as a diagnostic reference. Although this enables qualitative comparison of system predictions with manually derived solutions, manual judgments may themselves be subject to interpretation and do not constitute official ground-truth annotations.

Third, the validation and hidden-test configurations use different transcription and localization models in response to observable differences in document quality and identifier formatting. The current experiments do not isolate the contribution of each adaptation independently. Controlled component-level ablations would be required to quantify the effects of visual transcription, passage localization, identifier recovery, and verification separately.

Fourth, the final pipeline relies on several proprietary reasoning models. Exact reproduction may therefore depend on continued access to the corresponding model versions, provider interfaces, and inference configurations. Although the deterministic execution procedure and overall modular architecture are model-independent, changes to proprietary systems may affect exact numerical reproducibility.

Finally, the observed failure patterns are specific to the DocSem setting. Datasets with cleaner document images, standardized evidence identifiers, fewer semantically similar passages, or different evidence-evaluation protocols may exhibit a different balance between grounding and numerical reasoning errors. Further evaluation on additional document question answering benchmarks would be necessary to determine how broadly these findings generalize.

References

- Anthropic. 2026. [Introducing Claude Sonnet 5](#).
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. [Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks](#). *Transactions on Machine Learning Research*.
- Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. [FinQA: A dataset of numerical reasoning over financial data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- DeepSeek-AI. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [PAL: Program-aided language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.
- Google. 2026. [Gemini 3.1 pro: A smarter model for your most complex tasks](#).
- Yupang Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. [LayoutLMv3: Pre-training for document AI with unified text and image masking](#). In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091. Association for Computing Machinery.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. [OCR-free document understanding transformer](#). In *Computer Vision – ECCV 2022*, pages 498–517. Springer.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. [DocVQA: A dataset for VQA on document images](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.
- Moonshot AI. 2026. [Kimi k2.5: Visual agentic intelligence](#).
- OpenAI. 2026. [GPT-5.6: Frontier intelligence that scales with your ambition](#).
- PaddlePaddle. 2026. [PP-OCRv6: Universal multi-language text recognition](#).
- Qwen Team. 2026a. [Qwen3.5: Towards native multi-modal agents](#).
- Qwen Team. 2026b. [Qwen3.7-plus](#).
- Qwen Team. 2026c. [Qwen3.8-max](#).
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. [Large language models can be easily distracted by irrelevant context](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.
- Jyotika Singh, Fang Tu, Aziza Mirsaidova, Amit Agarwal, Hitesh Laxmichand Patel, Sandip Ghoshal, Miguel Ballesteros, Karan Dua, Yassine Benajiba, Weiyi Sun, Tao Sheng, Graham Horwood, Sujith Ravi, and Dan Roth. 2026. [Gsm-sem: Benchmark and framework for generating semantically variant augmentations](#).
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- xAI. 2026. [Grok 4.6](#).
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Model and Prompt Details

A.1 Model Configurations

Table 4 summarizes the models used in the validation experiments and final hidden-test pipeline.

The strongest validation configuration uses Qwen3.8-Max and GPT-5.6 Sol with Gemini 3.1 Pro for disagreement adjudication. The final test system uses GPT-5.6 Sol and Gemini 3.1 Pro as independent solvers, with Claude Sonnet 5 as the adjudicator.

B Prompt Templates

This appendix provides the prompt templates used for the main model-based components of our pipeline. Dynamic inputs are represented using braces.

B.1 Evidence Localization Prompt

Evidence Localization Prompt

You are given a user query and a document divided into blocks with valid block identifiers. Identify the smallest set of document blocks containing the complete quantitative problem relevant to the user query.

The user query may describe the desired information indirectly and may overlap lexically with unrelated passages. Treat it as a semantic retrieval cue rather than selecting evidence based only on shared words.

Do not solve the quantitative problem.

Select only the evidence required to reconstruct and solve the relevant problem. Every selected identifier must exactly match one of the provided valid identifiers.

Also extract the mathematical question stated in the selected evidence.

User query: {user_query}

Valid identifiers: {valid_ids}

Document blocks: {document_blocks}

Return the selected evidence identifiers and extracted mathematical question in the required structured format.

B.2 Quantitative Reasoning Prompt

Quantitative Reasoning Prompt

Solve the quantitative problem using only the provided evidence.

Identify the requested quantity, all relevant numerical values and units, and the mathematical relationships required to compute the answer.

Before returning the final result, verify that:

- all required quantities are included;
- irrelevant quantities are excluded;
- units are consistent or correctly converted;
- percentages and percentage-point differences are distinguished;
- the direction of comparisons is correct;
- repeated or grouped quantities are handled correctly; and
- every quantity used in the calculation is supported by the evidence.

Original user query: {user_query}

Mathematical question: {target_question}

Evidence: {evidence}

Return the result in the required structured format together with an explicit Python-compatible arithmetic expression.

B.3 Visual Transcription Prompt

Visual Transcription Prompt

Transcribe all visible document content faithfully.

Preserve the text exactly as visible, including capitalization, numerical values, punctuation, colons, line breaks, and block identifiers.

Do not summarize, paraphrase, explain, repair, or normalize the document text.

In particular, do not normalize visually similar characters in block identifiers. Reproduce each identifier exactly as it appears in the document.

Do not add Markdown formatting or commentary.

Return only the document transcription.

B.4 Evidence Identifier Correction Prompt

Evidence Identifier Correction Prompt

One or more selected evidence identifiers are invalid.

Select the supporting evidence again using only the valid identifiers provided below. Copy identifiers exactly as provided and do not modify, normalize, shorten, or reconstruct them.

The selected evidence must contain the complete quantitative problem relevant to the user query.

Valid identifiers: {valid_ids}

User query: {user_query}

Document blocks: {document_blocks}

Return valid evidence identifiers together with the extracted mathematical question in the required structured format.

B.5 Disagreement Adjudication Prompt

Disagreement Adjudication Prompt

Independently verify the quantitative problem using the provided evidence.

Do not select an answer based on the identity of a solver or its aggregate performance. Reconstruct the required computation directly from the evidence.

Check the requested quantity, relevant numerical values, units, calculation direction, and arithmetic in both candidate solutions.

If one candidate is correct, return that answer. If neither candidate is correct, independently derive the correct answer from the evidence.

Mathematical question: {target_question}

Evidence: {evidence}

Candidate solution 1: {candidate_1}

Candidate solution 2: {candidate_2}

Return the verified final answer in the required structured format.

Stage	Model / Configuration	Role
Validation OCR	PaddleOCR (PP-OCRv6) (PaddlePaddle, 2026)	Document transcription
Validation localization	Qwen2.5-7B-Instruct (Yang et al., 2024)	Evidence localization
Validation reasoning	Grok 4.6 (Fast) (xAI, 2026)	Quantitative reasoning
Validation reasoning	Gemini 3.1 Pro (Google, 2026)	Quantitative reasoning
Validation reasoning	Qwen3.7-Plus (Qwen Team, 2026b)	Quantitative reasoning
Validation reasoning	Qwen3.8-Max (Qwen Team, 2026c)	Quantitative reasoning
Validation reasoning	DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025)	Quantitative reasoning
Validation reasoning	DeepSeek-R1 (671B) (DeepSeek-AI, 2025)	Quantitative reasoning
Validation reasoning	GPT-5.6 Sol (OpenAI, 2026)	Quantitative reasoning
Validation reasoning	Claude Sonnet 5 (Anthropic, 2026)	Quantitative reasoning
Validation reasoning	Kimi K2.5 (Instant) (Moonshot AI, 2026)	Quantitative reasoning
Validation ensemble	Qwen3.8-Max + GPT-5.6 Sol	Independent solvers
Validation adjudication	Gemini 3.1 Pro	Disagreement adjudication
Test transcription	Qwen3.5-2B (Qwen Team, 2026a)	Visual transcription
Test localization	Qwen3.5-27B (Qwen Team, 2026a)	Evidence localization
Test reasoning	GPT-5.6 Sol + Gemini 3.1 Pro	Independent solvers
Test adjudication	Claude Sonnet 5	Disagreement adjudication

Table 4: Models used in the validation experiments and final hidden-test pipeline. Citations refer to the corresponding model release, model card, or official provider documentation.

B.6 Structured-Output Recovery Prompt

Structured-Output Recovery Prompt

Your previous response could not be parsed or contained an invalid final expression.
Re-evaluate the provided evidence and return a complete response in the required structured format.
Do not omit any required field. The arithmetic expression must be complete, numerical, and executable by the restricted Python evaluator.
Return only the corrected structured response.